KENYON & KENYON
LLP
Intellectual Property Law

RECEIVED
CENTRAL FAX CENTER

DEC 1 8 2006

333 W. San Carlos Street
Suite 600
San Jose, CA 95110-2731
408.975.7500
Fax 408.975.7501

Fax Transmission

From:	**Sumit Bhattacharya**	Date:	December 18, 2006
Direct Dial:	408.975.7950	Fax:	408.975.7501
Docket Number:	2207/9800	Total number of pages: (including cover)	18

Please deliver to:

Name	Company	Fax	Phone
M/S APPEAL BRIEFS - PATENTS	**U.S. Patent and Trademark Office**	**571.273.8300**	

Message:

Application No.	:	09/708,722	Confirmation No. 2194
Applicant	:	Stephen J. JOURDAN et al.	
Filed	:	November 9, 2000	
Title	:	INSTRUCTION SEGMENT RECORDING SCHEME	
TC/A.U.	:	2183	
Examiner	:	Aimee J. LI	

PAPER(s) ENTITLED:	Fee Transmittal (plus 1 copy)	2 pages
	Appeal Brief	15 pages

74734.1

[X] Original will not follow   [ ] Original will follow by   [ ] Regular Mail   [ ] Overnight Delivery   [ ] Hand Delivery

New York   Washington, DC   Silicon Valley   www.kenyon.com

RECEIVED
CENTRAL FAX CENTER

DEC 18 2006

PTO/SB/17 (10-04V2)
Approved for use through 07/31/2008. OMB 0851-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL
for FY 2006

Effective 10/01/2004. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($)	500.00

Complete if Known	
Application Number	09/708,722
Filing Date	November 9, 2000
First Named Inventor	Stephen J. JOURDAN et al.
Examiner Name	Aimee J. LI
Art Unit	2183
Attorney Docket No.	2207/9800

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: **11-0600**

Deposit Account Name: **Kenyon & Kenyon LLP**

The Director is authorized to: (check all that apply)
☒ Charge fee(s) indicated below ☒ Credit any overpayments
☒ Charge any additional fee(s) or any underpayment of fee(s)
☒ Charge fee(s) indicated below, except for the filing fee
to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	Fee Description	Fee Paid
1001	790	2001	395	Utility filing fee	
1002	350	2002	175	Design filing fee	
1003	550	2003	275	Plant filing fee	
1004	790	2004	395	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	
				SUBTOTAL (1)	($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

	Extra Claims	Fee from below	Fee Paid
Total Claims	__ - __ = __	X 50.00 =	
Independent Claims	__ - __ = __	X 200.00 =	
Multiple Dependent		X __ =	

Large Entity		Small Entity		
Fee Code	Fee ($)	Fee Code	Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple dependent claim, if not paid
1204	200	2204	100	** Reissue independent claims over original patent
1205	50	2205	25	** Reissue claims in excess of 20 and over original patent
				SUBTOTAL (2) ($)

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for ex parte reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	500.00
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	500	2452	250	Petition to revive – unavoidable	
1453	1,500	2453	750	Petition to revive – unintentional	
1501	1,400	2501	685	Utility issue fee (or reissue)	
1502	490	2502	245	Design issue fee	
1503	660	2503	330	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17 (q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	790	2809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	790	2801	395	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3)	($) 500.00

SUBMITTED BY

Name (Print/Type)	Sumit Bhattacharya	Registration No. (Attorney/Agent)	51,469	Telephone	(408) 975-7500
Signature				Date	December 16, 2006

Complete (if applicable)
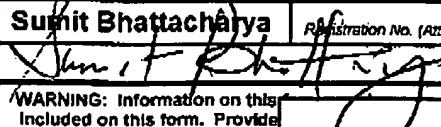
WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.
If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this paper is being facsimile transmitted to the USPTO, Fax No. 571.273.8300 on December 16, 2006.

Pilar Rodriguez
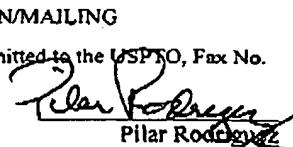
Patent   **Attorney Docket No.: Intel 2207/9800**
                                                  Assignee: Intel Corporation

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT	:	Stephen J. JOURDAN et al.	**RECEIVED**
			CENTRAL FAX CENTER
SERIAL NO.	:	09/708,722	**DEC 1 8 2006**
FILED	:	November 9, 2000	
FOR	:	INSTRUCTION SEGMENT RECORDING SCHEME	
GROUP ART UNIT	:	2183	
EXAMINER	:	Aimee J. LI	

M/S: APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

<table>
<tr><td colspan="2">CERTIFICATE OF MAILING</td></tr>
<tr><td colspan="2">I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office via facsimile number: (571) 273-8300 or deposited with the United States Postal Service as first class mail in an envelope addressed to: M/S: APPEAL BRIEF - PATENTS, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on</td></tr>
<tr><td>Dated: December 16, 2006</td><td>Pilar Rodriguez</td></tr>
</table>
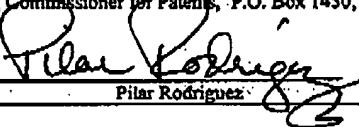
ATT   Board of Patent Appeals and Interferences

APPEAL BRIEF

Dear Sir:

This brief is in furtherance of the Notice of Appeal, filed in this case on October 16,

2006.

Application No. 09/708,722
Appeal Brief dated December 16, 2006

1. REAL PARTY IN INTEREST

The real party in interest in this matter is Intel Corporation. (Recorded November 9,

2000; Reel/Frame 011535/0333).

2. RELATED APPEALS AND INTERFERENCES

There are no related appeals.

3. STATUS OF THE CLAIMS

Claims 1-19 are pending in the application. Claims 1, 3-8, 12-15 and 17-19 are rejected

under 35 U.S.C. §103(a) as being unpatentable over Patel et al., *Improving Trace Cache*

Effectiveness with Branch Promotion and Trace Packing, in further view of Johnson, U.S. Patent

No. 5,924,092. Claims 2, 9-11 and 16 are rejected under 35 U.S.C. §103(a) as being

unpatentable over Patel, in further view of Johnson, in further view of Peled et al., U.S. Patent

No. 6,076,144.

4. STATUS OF AMENDMENTS

The claims listed on page A-1 of the Appendix attached to this Appeal Brief reflect the

present status of the claims.

5. SUMMARY OF THE CLAIMED SUBJECT MATTER

The embodiment of claim 1 generally describes a cache comprising: a cache line to store

an instruction segment (*see* e.g., page 5, lines 4-7 – Figure 2, 210) further comprising a plurality

of instructions stored in sequential positions of cache line in reverse program order (*see* e.g.,

page 5, lines 23-24).

The embodiment of claim 5 generally describes a segment cache for a front-end system in

a processor (*see* e.g., page 5, lines 4-7 – Figure 2, 210), comprising a plurality of cache entries to

store instructions of instruction segments in reverse program order (*see* e.g., page 5, lines 21-24

– Figure 2, 210 and Figure 4, 440).

The embodiment of claim 8 generally describes a method comprising building an

instruction segment based on program flow (*see* e.g., page 5, lines 14-15), and storing

instructions of the instruction segment in a cache entry in reverse program order (*see* e.g., page 5,

lines 21-24 – Figure 2, 210 and Figure 4, 440).

The embodiment of claim 14 generally describes a processing engine, comprising: a front

end stage to build and store instruction segments (*see* e.g., page 4, lines 3-12 – Figure 2, 200),

instructions provided therein in reverse program order (*see* e.g., page 5, lines 21-24 – Figure 2,

210 and Figure 4, 440), and an execution unit in communication with the front end stage (*see*

e.g., page 4, lines 10, 17, 22 and 24).

FIG. 1 is a block diagram illustrating the process of program execution in a conventional

processor. FIG. 2 is a block diagram of a front end processing system according to an

embodiment of the present invention. FIG. 3 is a block diagram of a segment cache according to

an embodiment of the present invention. FIG. 4 illustrates a relationship between exemplary

segment instructions a cache bank according to the embodiments of the present invention.

Application No. 09/708,722
Appeal Brief dated December 16, 2006

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Are claims 1, 3-8, 12-15 and 17-19 unpatentable over Patel et al., *Improving Trace Cache Effectiveness with Branch Promotion and Trace* Packing, in further view of Johnson, U.S. Patent No. 5,924,092?

B. Are claims 2, 9-11 and 16 unpatentable over Patel et al., *Improving Trace Cache Effectiveness with Branch Promotion and Trace* Packing, in further view of Johnson, U.S. Patent No. 5,924,092, in further view of Peled et al., U.S. Patent No. 6,076,144?

7. ARGUMENT

A. Claims 1, 3-8, 12-15 and 17-19 are <u>not</u> unpatentable over Patel in further view of Johnson.

Applicants submit the cited references do not teach, suggest or disclose at least "[a] cache comprising: a cache line to store an instruction segment further comprising a plurality of instructions stored in sequential positions of the cache line in reverse program order" (*e.g.*, as described in claim 1)

The Examiner asserts that it would be obvious to modify the instruction segment of Patel with the teaching of Johnson in order to store instructions of an instruction trace in reverse order "so that the frequently accessed and modified head of the trace will be moved and modified fewer times so that performance is improved." *See* Office Action dated 6/16/2006, paragraph 6. Applicants respectfully disagree. Applicants submit in order to establish *prima facie* obviousness, there must be some suggestion or motivation to modify the reference or combine the reference teachings. For at least the following reasons, there is no such suggestion or motivation here.

95310.1 -4-

Patel discloses the improvement of fetch rates in trace caches by employing branch promotion and trace packing. Branch promotion removes the overhead resulting from dynamic branch prediction by applying static branch prediction to strongly biased branches. Trace packing packs as many instructions as possible into a pending trace so that more instructions segments may be fetched during a single fetch cycle. However, Patel neither teaches nor suggests that the fetch rates may be improved by reversing the order of the instructions in the traces. Applicants submit there would be no motivation to do so, since reversing instruction order would not appear to improve fetch rates given the teaching of Patel.

By definition, a trace is a sequence of dynamically executed instructions, which may originally reside in non-continuous portions of the program memory, starting with a single entry instruction and ending with multiple exit instructions. For a typical trace, the head of the trace, *i.e.*, the first instruction in a sequence, is followed by the next executable instruction in the sequence, then the next, and so on. If the Examiner's assertions (discussed above) are correct, the first instruction is accessed and modified more than the second, third, etc., instructions. This is contrary to the known operation of the typical trace.

Applicants submit it is unclear how accessing the first instruction in a trace more frequently than the second instruction, and accessing the second instruction more frequently than the third, and so on, would improve the performance of a trace (thereby eliminating any motivation to do so). Since the trace defines sequential instructions, which perform a particular operation, accessing the instructions in decreasing frequency would in no way advance the completion of the particular operation. Indeed, such access of the trace would hinder the completion, thereby defeating the purpose of the trace.

Application No. 09/708,722
Appeal Brief dated December 16, 2006

Moreover, it is unclear how modifying the first instruction in a trace more frequently than the second instruction, and modifying the second instruction more frequently than the third, and so on, would improve the performance of a trace. Again, since the trace defines sequential instructions, which perform a particular operation, modifying the instructions in decreasing frequency would in no way advance the completion of the particular operation. Indeed, such modification would result in a different operation, thereby, defeating the purpose of the trace. Therefore, the Examiner's asserted motivation for modifying Patel with Johnson does not apply.

Furthermore, even if the head of the trace could be more frequently accessed and modified, the Examiner has provided no explanation of how such would improve the fetch rates of the trace cache of Patel.

As stated previously, there is no motivation to reverse the instructions in a Patel trace. Patel discloses using branch promotion to improve cache fetch rates. The purpose of branch promotion is to reduce the dynamic branching of strongly biased traces by applying static branches (or predictions). Applicants submit reversing the instructions so that the first instruction is listed last in the trace does not improve the branch promotion technique. Reversing the instructions does not reduce the dynamic branching of strongly biased traces. Moreover, it does not improve the fetch rates. As such, there is no reason a person of ordinary skill in the art would be motivated to reverse the instructions in a trace, while using branch promotion, to improve fetch rates.

Patel also discloses using trace packing to improve cache fetch rates. The purpose of trace packing is to increase the number of instructions fetched per fetch cycle. However, Applicants submit reversing the instructions so that the first instruction is listed last in the trace does not improve the trace packing technique. Moreover, such does not appear to improve the

95310.1   -6-

Application No. 09/708,722
Appeal Brief dated December 16, 2006

fetch rates. As such, a person of ordinary skill in the art would not be motivated to reverse the instructions in a trace, while using trace packing, to improve fetch rates.

Applicants further respectfully disagree with the Examiner's assertion that Johnson has taught that the static sorting algorithm stores elements in reverse order, since it stands to reason that the elements first added to the array would be accessed and used before the elements most recently added to the array. *See* Office Action dated 12/2/2005, page 11, paragraph 43, lines 1-3. Applicants disagree with the assertion and the associated rationales are found in paragraph 43. Applicants note the Examiner does not cite to any specific section of Johnson to support its assertion. *See* Office Action dated 6/16/2006, paragraph 40. Therefore, Applicants submit the claim that Johnson provides a motivation to combine its teaching with Patel is unsupported and erroneous.

However, previously in discussing Johnson, the Examiner cited column 4, lines 13-24. Column 4, lines 13-24 of Johnson state:

> For the illustrated embodiment discussed below, a static sorting algorithm is used which arranges the logical blocks of data in a logical page in reverse order, thereby placing the last logical block at the beginning of the array, and the first logical block at the end. Consequently, *the more frequent modifications* to the data in the first logical block require recompression and/or moving of fewer, or no other, subsequent frames than the *less frequent modifications* to the data in the last logical block of a page. In general, this results in a lower average number of *updated frames per data modification*, and thus improved overall performance. (*emphasis added*)

The cited section discusses improving overall performance based upon lowering the average number of update frames *per modification* in a data array. The cited section does not, however, discuss improving overall performance during *accessing* or *using* data in a data array. *Accessing* data or *using* data is not the same as *modifying* data. Modifying may be characterized as writing new data over old data in a data array (i.e., changing the data array), while accessing or using

95310.1   -7-

data may be characterized as loading and utilizing the data in the data array in its current state. The two are distinct concepts. The cited section of Johnson does not discuss the benefits of its sorting algorithm during data access or data "use" at all. To argue that "modifying" data, allegedly necessitates and therefore is the equivalent of "accessing" or "using" data is to eviscerate the meaning of concepts and functions readily known to those of ordinary skill in the art as being separate and distinct. *See* Office Action dated 6/16/2006, paragraph 48.

In fact, the Johnson reference is not directed toward improving performance during data access or use at all, but rather limited to discussing improving performance during data modification. Nevertheless, the Examiner asserts:

> Johnson has taught that the static sorting algorithm stores elements in reverse order, since it stands to reason that the elements first added *to the array would be accessed and used* before the elements most recently added to the array. (*emphasis added*) *See* Office Action, paragraph 43, lines 1-3.
> *and*
> Johnson has taught that entries that were placed at the beginning of an array, *e.g.* placed first within the array, *are more likely to be accessed first*, so, by placing these elements at the end of an array where there are less elements dependent on that one particular element to be made when the elements are modified, such as when Patel's instruction segments are fetched in split groups, less time is required to update the element. (*emphasis added*) *See* Office Action, paragraph 43, lines 14-18.

Applicants submit that these overbroad assertions are erroneous and unsupported by the Johnson reference. Applicants submit the related assertions based upon these interpretations of Johnson found in paragraph 43 of the Examiner are erroneous as well, and insufficient to support a proper teaching, suggestion or motivation to combine the teachings of Patel and Johnson.

The Examiner counters by citing to column 4, lines 27-29 in Johnson ("...it should be appreciated that the elements may represent practically any types of memory blocks or segments, having any fixed or variable size.") *See* Office Action dated 6/16/2006, paragraph 42. A generalized statement of Johnson's ability to apply its system to larger memory blocks of fixed

95310.1　　　　　　　　　　　　　　-8-

Application No. 09/708,722
Appeal Brief dated December 16, 2006

or variable size is not a sufficient motivation to combine when considering, as shown above, the teachings of Johnson are functionally incompatible with those of Patel. To argue that despite the fact the Johnson reference teaches away from combining with the Patel reference, one would still be motivated to combine them in such a manner is unsupported and inadequate.

The Examiner further asserts the last sentence of the cited section column 4, lines 13-24 provide the necessary motivation to combine the teachings of Patel and Johnson (citing specifically "...In general, this results in a lower average number of updated frames per data modification, and thus improved overall performance.") *See* Office Action dated 6/16/2006, paragraph 38. Applicant disagree, and submit a generalized statement directed toward a purported "improvement" on the prior art is not sufficient to supply a *motivation to combine*. Otherwise every patent application purporting to improve upon the prior art may on its face arguably offers a motivation to combine. Clearly this is not the case.

Applicants maintain that a person of ordinary skill in the art would not be motivated to reverse the instructions in a trace, while using trace packing, to improve fetch rates and that as such, claim 1 is allowable in its present form. Independent claims 5, 8 and 14 contain similar allowable limitations, and are therefore allowable for similar reasons. Claims 2-4, 7, 9-13 and 15-19 are allowable for depending from allowable base claims.

B.   Claims 2, 9-11 and 16 are not unpatentable over Patel, Johnson and in further view of Peled.

The deficiencies are not corrected by Peled. Peled discloses a cache organized around trace segments of the running programs rather than an organization based on memory addresses. However, Peled fails to provide any motivation for modifying Patel with Johnson. Moreover,

95310.1   -9-

Application No. 09/708,722
Appeal Brief dated December 16, 2006

there is no motivation disclosed to modify Patel with Johnson and Peled to arrive at the claimed

invention. Accordingly, the Examiner has failed to establish a *prima facie* case of obviousness

over Patel in view of Johnson in further view of Peled.

CONCLUSION

For at least these reasons, the Claims 1-19 are believed to be patentable over the cited

references, individually and in combination. Withdrawal of the rejections is, therefore,

respectfully requested.

Appellant therefore respectfully requests that the Board of Patent Appeals and

Interferences reverse the Examiner's decision rejecting claims 1-19 and direct the Examiner to

pass the case to issue.

The Examiner is hereby authorized to charge any additional fees which may be necessary

for consideration of this paper to Kenyon & Kenyon Deposit Account No.

11-0600.

Respectfully submitted,

KENYON & KENYON LLP
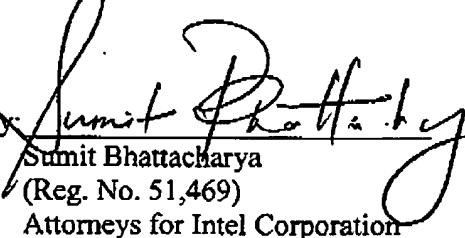
By: Sumit Bhattacharya
(Reg. No. 51,469)
Attorneys for Intel Corporation

Date: December 16, 2006

KENYON & KENYON LLP
333 West San Carlos St., Suite 600
San Jose, CA 95110
Telephone: (408) 975-7500
Facsimile: (408) 975-7501

95310.1   -10-

APPENDIX
(Brief of Appellants Stephen J. Jourdan et al.
U.S. Patent Application Serial No. 09/708,722)
8. CLAIMS ON APPEAL

1.   A cache comprising:

a cache line to store an instruction segment further comprising a plurality of instructions

stored in sequential positions of cache line in reverse program order.

2.   The cache of claim 1, wherein the instruction segment is an extended block.

3.   The cache of claim 1, wherein the instruction segment is a trace.

4.   The cache of claim 1, wherein the instruction segment is a basic block.

5.   A segment cache for a front-end system in a processor, comprising a plurality of cache

entries to store instructions of instruction segments in reverse program order.

6.   Apparatus comprising:

an instruction cache system,

an instruction segment system, comprising:

a fill unit provided in communication with the instruction cache system,

the segment cache of claim 5 included therein, and

a selector coupled to an output of the instruction cache system and to an output of the segment

cache.

7.   Apparatus of claim 6, wherein the instruction segment system further comprises a

segment predictor provided in communication with the segment cache.

8.   A method comprising:

building an instruction segment based on program flow, and

storing instructions of the instruction segment in a cache entry in reverse program order.

9.   The method of claim 8, further comprising:

building a second instruction segment based on program flow, and

if the first and second instruction segments overlap, extending the first instruction segment to

include non-overlapping instructions from the second instruction segment.

10.   The method of claim 9, wherein the extending comprises storing the non-overlapping

instructions in the cache in reverse program order in successive cache positions adjacent to the

instructions from the first instruction segment.

11.   The method of claim 8, wherein the instruction segment is an extended block.

12.   The method of claim 8, wherein the instruction segment is a trace.

13.   The method of claim 8, wherein the instruction segment is a basic block.

14.   A processing engine, comprising:

a front end stage to build and store instruction segments, instructions provided therein in reverse

program order, and

an execution unit in communication with the front end stage.

15.   The processing engine of claim 14, wherein the front-end stage comprises:

an instruction cache system,

an instruction segment system, comprising:

      a fill unit provided in communication with the instruction cache system,

      a segment cache, and

a selector coupled to an output of the instruction cache system and to an output of the segment

cache.

16.   The processing engine of claim 15, wherein the instruction segments are extended blocks.

17.   The processing engine of claim 15, wherein the instruction segments are traces.

18.   The processing engine of claim 15, wherein the instruction segments are basic blocks.

19.   The processing engine of claim 15, wherein the instruction segment cache system further

comprises a segment predictor provided in communication with the segment cache.

9.   **EVIDENCE APPENDIX**

No further evidence has been submitted with this Appeal Brief.

Applicatio No.: 09/708,722
Appeal Brief dated December 16, 2006

Attorney Docket No. 2207/9800

10. RELATED PROCEEDINGS APPENDIX

Per Section 2 above, there are no related proceedings to the present Appeal.

95310.1   -5-

PAGE 18/18 * RCVD AT 12/18/2006 7:17:21 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-3/7 * DNIS:2738300 * CSID:14089757501 * DURATION (mm-ss):05-32   P.18